A General Predictive Performance Model for Wavefront Algorithms on Clusters of SMPs

Adolfy Hoisie, Olaf Lubeck, Harvey Wasserman, Fabrizio Petrini, Hank Alme
Parallel Architectures and Performance, CIC-3
Los Alamos National Laboratory
Email: hoisie@lanl.gov

1. Introduction.

We have recently been studying the performance of wavefront algorithms implemented using message passing on 2-dimensional logical processor arrays [1,2]. Wavefront algorithms are ubiquitous in parallel computing, since they represent a means of enabling parallelism in computations that contain recurrences. Our particular interest in wavefront algorithms derives from their use in discrete ordinates neutral particle transport [3] computations, but other important uses are well known [4-7].

The basis of wavefront parallelism is the data dependence graph shown in Figure 1, in which the nodes may represent either physical grid points or logical processors. In the latter case, a computation progresses as a wavefront and "scans" through a processor grid with pairs of processors sending and receiving boundary data required in order to update a portion of the physical mesh. Those processors within each wavefront, i.e., those on a diagonal, are algorithmically independent. Intuitively, then, the nominal benefit of wavefront parallelism is related to the (continuously-changing) length of a diagonal. However, additional concurrency can be achieved by "blocking" the computation, resulting in more wavefront “sweeps” using smaller computational subgrids. This reduces processor idle time that accumulates as processors await their turn to compute, but requires that processors communicate more often. This tradeoff between processor utilization and communication requirements is characteristics of wavefront algorithms. An important task of performance models such as those described in [1,2] and the one proposed in this paper is to capture this tradeoff and the influence of the blocking parameters on the overall runtime of the application.

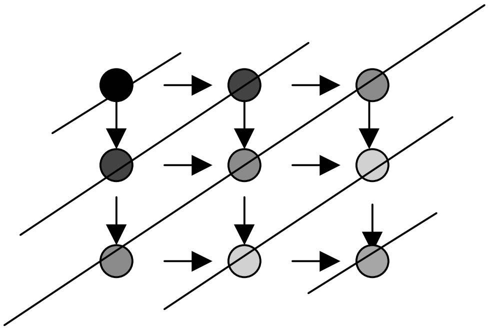

Figure 1. Schematic of wavefront parallelism

A key component of our work, then, has been to model performance of wavefront algorithms to predict overall performance as well as optimal blocking sizes given a machine's computation and communication parameters. In previous papers [1,2] we developed a closed-end, analytical model for the parallel performance of wavefront algorithms implemented on a specific class of parallel computers - those in which a logical processor mesh could be embedded into the machine topology such that each mesh node is mapped to a unique processor and each mesh edge is mapped to a unique router link. When this condition is met, there is a high level of message concurrency across the processor grid. We now refer to this model as the "MPP" case, since it accurately describes machines such as the CRAY T3E and the older IBM RS/6000-SP (without SMP nodes), both of which have congestion-free connectivity between any logically adjacent nodes. This model describes a pipelined series of wavefronts with a characteristic (and constant) pipeline length and repetition delay.

In this paper we concern ourselves with the generalization of this model when the network topology is not uniform, such as in a cluster of SMPs interconnected by a network of lower dimensionality. Here, a wavefront can arrive at an inter-SMP boundary and be delayed, because a message from a previous wavefront is already using needed links/wires between SMP hosts. The model should capture how this decreased connectivity affects the wavefront pipeline parameters (pipeline length and repetition delay), the message concurrency, and thus, overall performance, compared to the simpler MPP case. The work has immediate relevance to the DOE Accelerated Strategic Computing Initiative, which has embraced clustered SMP technology as the primary architecture used to build multi-TeraOp systems. We validate our new model using both simulation experiments and experimental data from LANL's cluster of SGI Origin2000.

2. Review of MPP Model and Basic Description of the SMP Cluster Case.

The point of departure for our model is a pipelined wavefront abstraction [1,2], in which N_{sweep} wavefronts scan the P_x X P_y processor grid, each requiring N_s steps, with a repetition delay of d between each wavefront. The total number of steps for all wavefronts is given by equation (1).

$$Steps = N_S + d(N_{sweep} - 1) \qquad (1)$$

The first wavefront exits the pipeline after N_s stages and subsequent waves exit at the rate of $1/d$. The challenge is to find N_s and d for both computation and communication. In reference [1] we showed that these are captured completely in equations (2) and (3).

$$T^{comp} = [(P_x + P_y - 1) + (N_{sweep} - 1)] * T_{cpu} \qquad (2)$$

$$T^{comm} = [2(P_x + P_y - 2) + 4(N_{sweep} - 1)]*T_{msg} \qquad (3)$$

Thus, the number of steps in the computation pipeline is simply the number of diagonals in the processor array. The cost of each step, T^{comp}, is a function of the number of grid points per processor ("subgrid") and some characteristic floating-point computation rate, Rflops. The repetition delay for computation, d^{comp}, is 1 (i.e., the time for completing one diagonal in the sweep). The cost of any single communication stage is the time of a one-way, nearest neighbor communication. This time, for a message of length N_{msg}, is given by:

$$T_{msg} = t_0 + \frac{N_{msg}}{B} \qquad (4)$$

where t_0 is the message startup time and B is bandwidth.

A key element of the communication model is that the repetition delay between communication pipelines is 4, because, as shown in Figure 2, a message sent from any processor (say processor 0) to its east neighbor (processor 1) on the second sweep cannot be initiated until processor 1 completes its communication with its south neighbor (processor 3) from the first sweep.

For this model, we assume three things: (1) blocking synchronous communications; (2) messages initiated by the same processor occur sequentially in time and messages must be received in the same order that they are sent; (3) as implemented, the order of receives is first from the west, then from the north, and the order of sends is first to the east and then to the south. However, there is no loss of generality by making these assumptions; i.e., the algorithm is "self-synchronizing," so the use of blocking send/receives does not matter, and changing the order of sends/receives leads to the same concurrency (and number of steps) for communications.

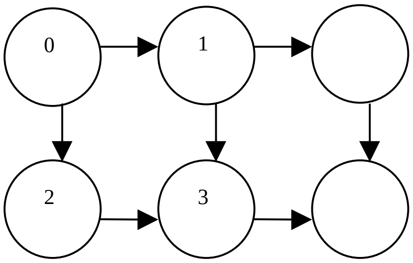

Figure 2. Repetition delay for the communication pipeline in the MPP case

When the network topology is not uniform, as in a case of a cluster of SMPs interconnected by a network of lower dimensionality, disruption in the wavefront pipeline may occur so that otherwise-independent wavefronts may "collide", i.e. wavefronts competing for a link are delayed. The abstraction we use to describe wavefronts on clusters of SMPs is a "pipeline with bottlenecks". When pipelined wavefronts are delayed at a bottleneck (the inter-SMP boundary), subsequent wavefronts may be delayed, too. This delay can alter the frequency of the pipeline, and can propagate back up to the processor that initiated the wavefront. We have found that this back-propagation takes place during some transitional number of wavefronts. A steady-state is then reached in which wavefronts may scan the processor array at a slower rate compared with the MPP case and with a variable and periodic frequency. We now provide a rigorous, quantitative analysis of this process, the object of which is a modified version of equations (2) and (3), giving the number of steps required to scan the bottlenecked wavefront pipeline. We anticipate that the geometry of the SMPs and the specifics of the inter-SMPs connectivity will ultimately dictate the parameters of the pipeline. Note, we deliberately use the conditional "may" in the previous sentences. An important part of this work is to discover the conditions under which performance reverts to the original MPP case.

3. Complexity Analysis. Model Development.

We now consider a logical *m* X *n* cluster of SMP hosts, each of which is a logical *Sx* by *Sy* system with full connectivity. Note that this representation is the logical view of the processor configuration "seen" by our discrete ordinates particle transport application, which uses a 2-D processor domain decomposition. The SMP hosts are linked to one another via a connection scheme that allows L_x concurrent messages to pass in the *x* direction and L_y concurrent messages to pass in the *y* direction. Lower-dimensional connectivity implies that L_x or $L_y < \max(S_x, S_y)$. This is schematically depicted in Figure 3 for a 2 X 2 cluster, with L_x=2 and L_y=4. Throughout this paper, processors are counted contiguously. For example, the upper-left processor of the (2,1) SMP in Figure 3 is numbered (1, S_y+1) and the corner processor of the (2,2) SMP is numbered (S_x+1, S_y+1). When the number of links in each direction is equal, we use $L = L_x = L_y$.

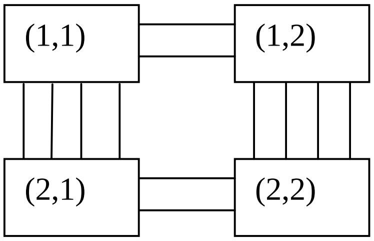

Figure 3. Logical representation of a 2X2 cluster of SMPs

In order to simplify presentation, we will use the following intuitive facts:

a) Collisions take place in both directions (*x* and *y*). However, due to the interweaving of the *x*- and *y*-direction communication steps, analysis of collisions in the *y* direction (which ends later than the x direction due to the posting first of the east-west message receive) is sufficient for deriving the formula for the overall number of communication steps.

b) Collisions affecting the communication pipeline take place only on the first inter-SMP link in either direction. Corollary: if a collision didn't take place in the first communication step in which that wave is involved at an inter-SMP boundary, that wave will be collision-free for the rest of its passage through the processor array.

c) The collision pattern (and implicitly the wavefront dynamics) is dictated by the first SMP (and its boundaries) the wave scans. The waves will then move unimpeded through all the other SMPs and their boundaries. This is a direct consequence of the pipelining in the pipeline with bottlenecks model.

We also assume that waves cannot collide back; i.e. waves cannot be influenced by subsequent waves. This assumption does not change the generality of the analysis, and we will return to this statement for complete clarification.

Our development of the model proceeds by examining the cases where collisions occur for the first time, by induction. We begin by modifying equation (1), which gives the number of communication steps for N_{sweep} wavefronts to scan the grid, so that it instead gives the number of communication steps for the I^{th} wavefront to scan the first SMP:

$$N_{steps}=2(P_x-1)+2(P_y-1)+4(I-1) \qquad (5)$$

The first inter-SMP boundary in the *y*-direction is between processors S_y and S_y+1. Concentrating now on the *y*-boundary between SMPs, we note that equation (5) suggests that all communications in the *y*-direction take place on even-numbered timesteps. The first wavefront will move unimpeded through the entire processor grid, with all its communication steps across the boundary and elsewhere in the *y*-direction being “even-labeled” (even numbers). The second wavefront will be collision-free

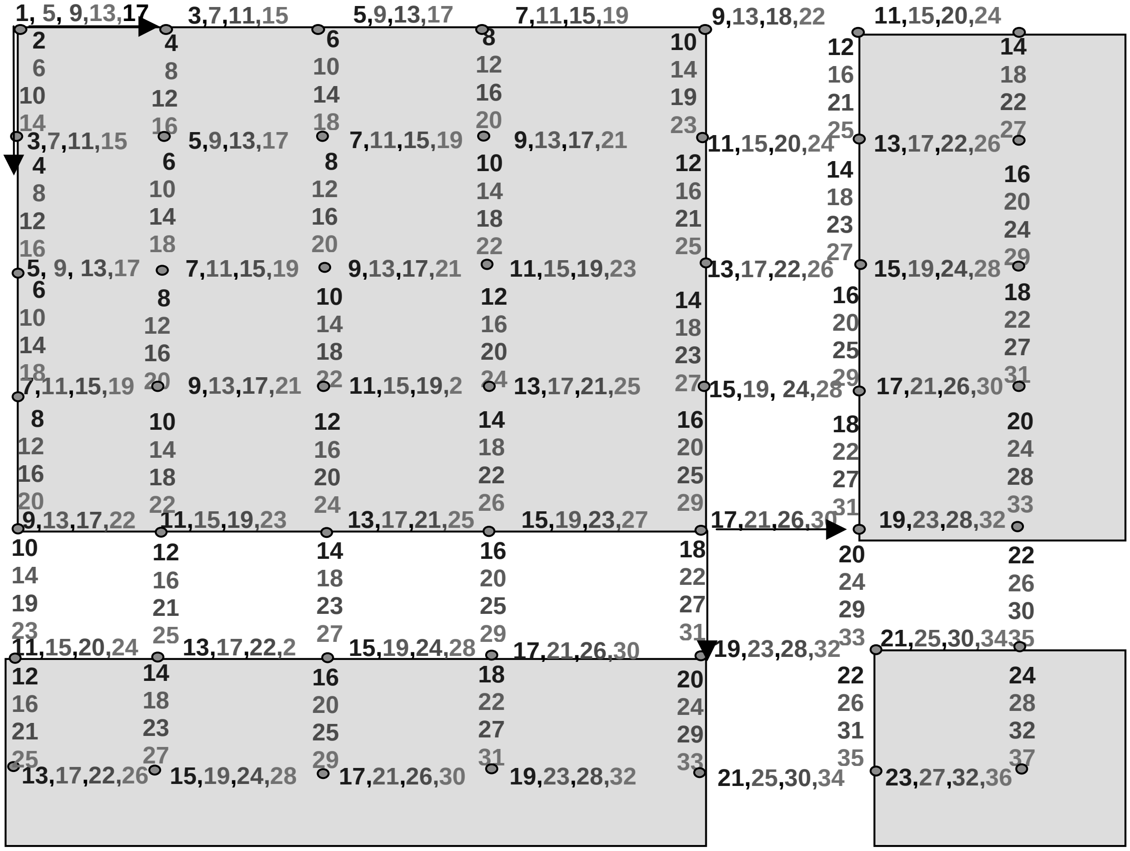

Figure 4. Emulation of the communication steps on 4 wavefronts among 4 clusters of 4 wavefronts among 4 clusters of SMPs each containing 5 X 5 processors. Two links in each direction are assumed between the SMPs. The gray areas show the SMPs, the white area is the inter-SMPs boundary. The wavefronts move from the upper left corner to the lower right corner of the processor array. The numbers shown represent the communication steps at which messages are exchanged between those processors.

provided that $L > 1$. In general, the first L waves will be collision-free, with the number of communication steps required for wavefront I to reach processor (S_x+1, S_y+1) given by:

$$S_I = 2S_y + 2S_x + 4(I-1) \qquad I <= L \qquad (6)$$

This can be seen in Figure 4 obtained by emulation: the first 2 waves are collision-free when L=2, as shown by the delay of 4 between these waves at the inter-SMP boundaries.

Wavefront number L+1 will collide, provided that:

$$2S_y+4(L+1-1) <= 2S_y+4(1-1) +2S_x \qquad (6a)$$

stating that the timestep on the first link in wave L+1 needs to be at most equal to the timestep in the first wave at the end of the boundary for a collision to occur. This is shown in figure 4 by the third wavefront, which would communicate over the first inter-SMP link in the y-direction at timestep 18. However, given the availability of two links only, this third wave collides, as the two links are taken by the first and second wavefront communicating at timestep 18 on the 5t^{h} and 3rd link respectively.

(6a) leads to:

$$S_x >= 2L \qquad (7)$$

Equation (7) is revealing, because it shows that for our algorithm full connectivity is *not* required for wavefronts to scan unimpeded. In fact, all that is required is that there be at least half as many links as there are processors on the (logical) SMP boundary, in which case the analysis trivially reduces to the MPP case.

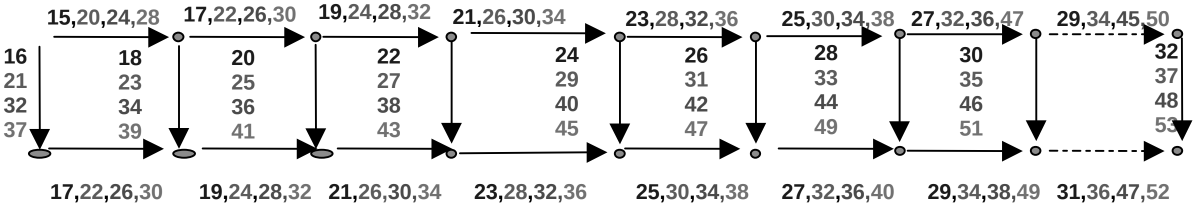

Figure 5. The inter-SMP boundary in the y-direction for 8 X 8 processors within the SMP. The dotted lines arrows show the inter-SMP boundary in the x-direction.

We now consider those cases in which condition (7) is met and collisions occur. Now, the timestep at which wavefront L+1 crosses the inter-SMP boundary is bumped up by one. This changes the parity of timesteps in the y-direction, meaning that wave L+1 will cross the boundary on an odd timestep. Recall that the first L wavefronts are collision-free and of even parity. The next L wavefronts will necessarily preserve the odd parity achieved by wavefront L+1, because the differential between two consecutive waves is always equal to 4. For this group of waves with odd parity, the number of communication steps to reach processor (S_x+1, S_y+1) is given by:

$$S_I = 2S_y + 2S_x + 4(I-1) + 1,\ L+1 <= I <= 2L \quad (8)$$

This is illustrated in Figure 4 by waves number 3 and 4, which switch from even to odd timesteps when communicating across the SMPs in the y-direction.

To continue the discussion we assume for the moment that $S_x > S_y$. We'll comment on this restriction at the end of the chapter.

The third group of waves, beginning with wavefront 2L+1, will be bumped up again, because at this point, the L waves from the second group are still utilizing the links. This third group of L waves will switch back to even parity. The condition:

$$2S_y+4(2L+1-1)+1+1 <= 2S_y+2S_x+4(1-1), \quad (9)$$

checking if the timestep on the first link at the inter-SMP boundary, as given by equation (8) and incremented by 1 to revert to even-labeled timesteps, is lower than or equal to the timestep of the first wavefront in the first group of (even-labeled) waves on the first link out of the SMP, leads to

$$S_x >= 4L+1 \quad (10)$$

Where (10) is not satisfied, i.e. collisions with the first group of even-labeled waves do not occur, the total number of steps is given by:

$$S_I = 2S_y+2S_x+4(I-1)+1+1 \quad 2L+1 <= I <= 3L \quad (11)$$

By induction we conclude that all subsequent groups of L waves will alternate parity in a similar fashion, with the total number of steps given by:

$$S_I = 2S_y+2S_x+4(I-1) + [\mathrm{int}((I-1)/L)] \quad \text{with } I=1,N_{sweeps} \quad (12)$$

These formulae apply for the waves depicted in Figure 4, because equation (10) is not met when $S_x = S_y = 5$ and $L = 2$.

If (10) is satisfied, as in the case depicted in Figure 5, then the third group of L waves will begin on an even time step that is equal to the time step at which the first wave in the previous group of the same parity (in this case wave 1) reached the first *intra*-SMP link.

In Figure 5, this is shown by the third wavefront communicating over the first link at timestep 32, as wavefronts 1 and 2 communicate on all timesteps between 25 and 31 over the single inter-SMP link available. Note that Figure 5 only depicts the y-boundary between two clusters of SMPs having 8 X 8 processors each, with L=1.

The timestep on which this third group of L waves will end on is given by

$$S_I = 2S_y+2S_x+2S_x \quad \text{with } 2L+1 <= I <= 3L \quad (13)$$

Generally, when (10) is satisfied, the first wave of each even group ends at:

$$S_I = 2S_y + 2S_x + [\mathrm{int}((I-1)/L)]S_x \quad \text{with } I=1,2L+1,4L+1\ldots \quad (14)$$

and the first wave of each odd group ends at:

$$S_I = 2S_y + 2S_x + 4(L-1) + 5 + [\mathrm{int}((I-1)/\ L)-1]S_x \text{ with } I=L+1,3L+1,5L+1\ldots \qquad (15)$$

Obviously, the timestep for all the other *L*-1 waves in each group are obtained by adding 4 to the appropriate equation (14) or (15) for each wave in the group.

In summary, when equation (10) is false then equations (6), (8), and (13) are relevant. When equation (10) is true, then equations (6), (8), (14) and (15) apply.

The number of time steps needed for one wavefront to scan the entire cluster of SMPs is given by:

$$S_I + (m-2)*2S_x + (n-2)*2S_y + 2(S_x-1)+2(S_y-1). \quad (16)$$

If the cluster is unidimensional then (16) becomes:

$$S_I^{'} + (m-2)*2S_x + 2(S_x-1)+2(S_y-1) \quad \text{when } n=1 \qquad (17)$$

where $S_I^{'}$ is the appropriate S_I without the $2S_y$ term.

When *m* =1 then the number of communication steps is given by:

$$S_I^{'} + (n-2)*2S_y + 2(S_x-1)+2(S_y-1) \qquad (18)$$

where $S_I^{'}$ is the appropriate S_I without the $2S_x$ term.

If the number of links in the *x*-direction (L_x) is different than the number of links in the *y*-direction L_y , then $L = \min(L_x, L_y)$.

Previously, we anticipated that a steady-state regime for the movement of the wavefronts would be achieved after a transitional period. The number of groups of wavefronts in the transitional period is min (S_x,S_y) + 1. This is illustrated in Figure 6, where the number of wavefronts in each group is 1 (since L_x=1), and the number of transitional groups as seen in the figure is 4 (equal to S_x + 1). Only after 4 wavefronts the repetition delay between wavefronts becomes equal to the one at the inter-SMP boundary.

When L=S_y=S_x, then (3) is not satisfied and the entire analysis reduces to the MPP case described in [1,2] and summarized by equation (2).

The condition (10) was obtained for the case when $S_x > S_y$. In fact, generally the condition is: $\max(S_x , S_y) > 4* \min(L_x, L_y) + 1$ and the factor multiplying the *int* function in equations (16) and (17) is $\max(S_x , S_y)$ instead of S_x .

The characteristics of any pipeline model are apparent for this case of a pipeline with bottlenecks. We note that equations (12), (16) and (17) all contain two distinct parts: one independent of *I* and one dependent on *I*. The *I*-independent part represents the number of steps in each wavefront, while the *I*-dependent part represents the pipeline frequency and contains the total number of wavefronts. The legitimacy of the pipeline with bottleneck model proposed is now proven.

An interesting consequence, alluded to earlier, is that when steady-state is reached and (10) is also satisfied, the repetition delay that occurs when even-parity wavefronts follow odd-parity wavefronts will be different than the delay that occurs when odd-parity wavefronts follow even-parity wavefronts. Thus, the overall frequency of the pipelined wavefronts is variable and periodic. If (10) is not satisfied, then the repetition delay between groups of even and odd timesteps and groups of odd and even timesteps is constant and equal to 5 (from eqn (12)).

Finally, we assumed that wavefronts cannot collide back and claimed that the generality of the analysis is not affected by this. If the assumption were not true, the only consequence would be that wavefronts in a group of the same parity would no longer be contiguous, and would be interspersed with wavefronts belonging to a group of a different parity. The analysis would be greatly complicated, whereas the end result would be the same.

4. Experimental testbed

“Blue Mountain” at the Los Alamos National Laboratory is a cluster of 48 Origin 2000 SMPs each equipped with 128 processors, for a total of 6144 processors. The communication fabric utilized to connect these building blocks is made of HiPPI 800 (High Performance Parallel Interface) network interfaces and switches [7]. The network topology is designed in such a way that SMPs are directly interconnected.

The interconnection diagram for 6 SMPs is depicted in Figure 7. In the logical representation utilized in the previous chapter, and depicted in Figure 3, the interconnect in Figure 7 amounts to $L_x = L_y = 2$.

HiPPI network interfaces are unidirectional channels with a peak bandwidth of 100 MB/s. A logical bi-directional channel is set up by bundling two HiPPI channels together. On the

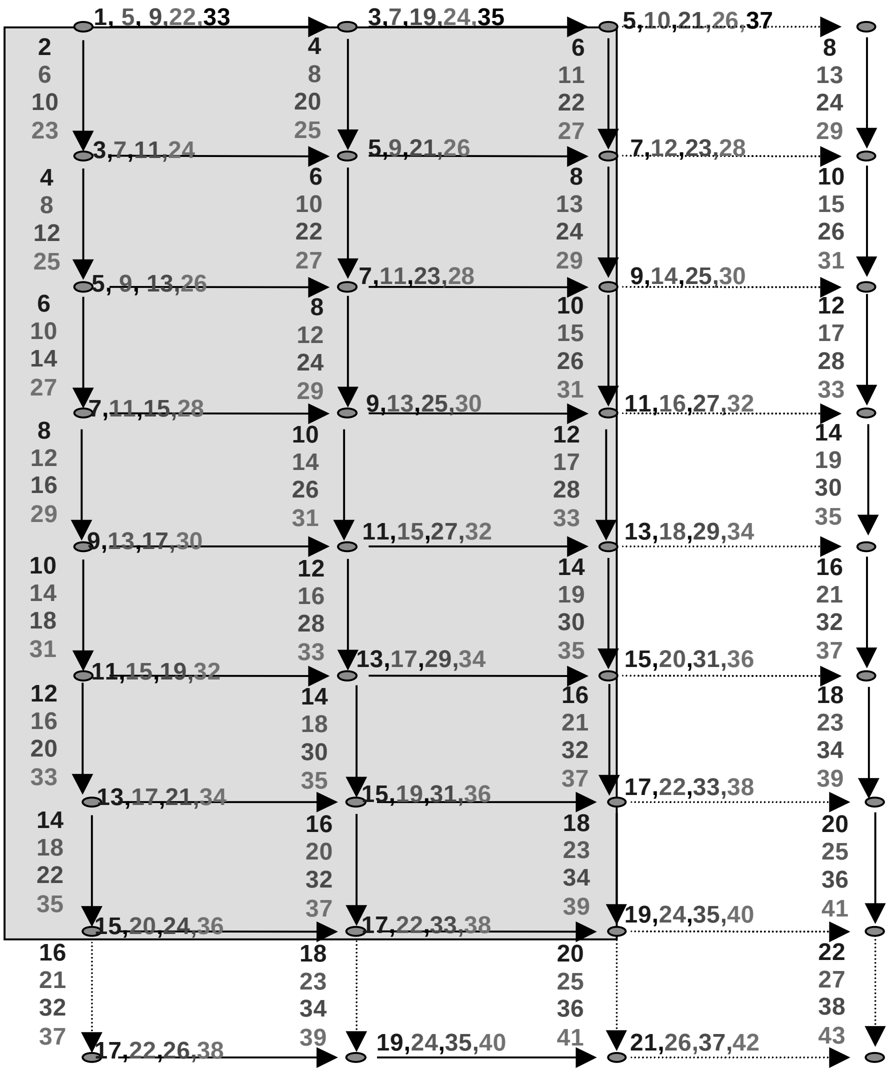

Figure 5. The inter-SMP boundary in the y-direction for 8 X 8 processors within the SMP. The dotted lines arrows show the inter-SMP boundary in the x-direction.

sending side, the HiPPI interface provides a direct memory access (DMA) read engine that can move data from the SMP to the HiPPI link. The interface is controlled by a MIPS R3000 processor. The receiving side has a symmetric layout, with a DMA write engine and a communication processor. The HiPPI interface provides a fixed number of virtual endpoints, 8 in the current implementation. Flow-control is performed at packet-level.

Application-level communication uses an implementation of MPI specifically tailored for HiPPI. MPI messages are packetized using a chunk size of 16KB. Messages can be striped over multiple HiPPI links (i.e., an MPI message larger than 16KB can be fragmented in packets and these packets may be sent over different HiPPI interfaces).

The MPI implementation allows three HiPPI channel allocation policies: deterministic, adaptive and round-robin. With the deterministic policy each message is always routed to the same HiPPI interface, while the adaptive policy picks the least loaded interface. The round robin policy distributes the traffic across the available interfaces using a user-defined order. In the rest of this paper we will consider the default policy, the adaptive one.

Due to the high cost of re-mapping the physical addresses on the HiPPI card (larger than 1 ms), and the limitations on addressing the whole physical memory on an SMP, the MPI implementation first copies the packet in a temporary buffer area, both on the sending and the receiving side. Each message is first copied from the user space to the first buffer on the sending side, then to the buffer on the receiving side and finally to the user space of the receiver.

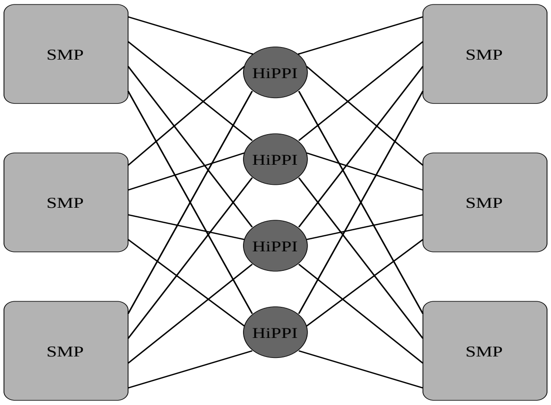

Figure 7. Connectivity diagram for 6 SMPs. Each packet can be sent from an SMP to any other SMP passing through a single HiPPI switch. Multiple paths are provided by different HiPPI switches

4.1 Communication Performance of the HiPPI Interfaces

To expose the communication characteristics of the MPI implementation and of the underlying hardware, we run a benchmark that analyzes the actual communication bandwidth as a function of the message size and the number of processes involved in the communication. The goal is to generate the “fingerprint” of communication over HiPPI.

The benchmark consists of two main loops. In the outer loop, we define two sets of processes of the same size, ranging from 2 to 128 processes. All of the processes in each set are bound to the same SMP. Each process in the first set sends messages to a partner process in the second set. The communication pattern generated by this benchmark is unidirectional, as is the one in a wavefront algorithm, where all processes in one SMP propagate unidirectional waves to the neighboring SMPs. In the inner loop of the benchmark we vary the communication granularity, i.e., the message size. The experimental results are shown in Figure 8. The graph shows the global bandwidth achieved by the collective communication pattern as a function of the message size and the number of pairs of processes involved.

Along the “Message size” axis, we can first identify a region, for messages smaller than 2048 bytes, where the communication pattern is largely dominated by the startup latency. This region is highlighted in the graph by the “Small message granularity” arrow.

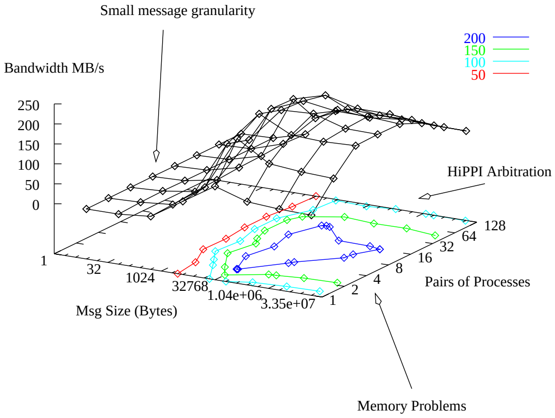

Figure 8. Inter-SMP communication performance over HiPPI links.

The second region is delimited by messages ranging from 32KB to approximately 1MB. In this region the global bandwidth increases linearly with the message size, up to a maximum of about 70 MB/s per each individual HiPPI link. For larger messages (above 1 MB) performance degrades, due to buffer memory allocation and coherence protocols problems. This region is identified by the “Memory problems” arrow. In interpreting the data along the “Pairs of Processes” axis, for a number of communicating pairs of processes larger than 16, the overhead introduced by the HiPPI arbitration protocol limits the communication throughput. For a small number of communicating pairs the available bandwidth is lower because the HiPPI communication protocol cannot efficiently stream a single packet over multiple links.

The overall best operating region is for messages of about 1 MB and between 8 and 16 pairs of processes.

5. Validation of the model

In this section we present experimental data to validate the proposed model for the performance of wavefront algorithms on clusters of SMPs.

The data presented was collected on the Origin 2000 cluster described in section 4. Our vehicle for these studies is a “compact application” called SWEEP3D [3,1], a time-independent, Cartesian-grid, single-group, “discrete ordinates” deterministic particle transport code taken from the DOE Accelerated Strategic Computing Initiative (ASCI) workload. SWEEP3D represents the core of a widely utilized method of solving the Boltzmann transport equation. Estimates are that deterministic particle transport accounts for 50-80% of the execution time of many realistic simulations on current DOE systems.

We are using a fixed subgrid size per processor in all the runs. Its size of 8 X 8 X 320 was obtained based on the 2D processor decomposition described in Chapter 2 and on an estimate of the largest problem size that can be computed on the full machine configuration described in Chapter 4. Given the subgrid size selected, the size of the messages is 38 Kbytes. From figure 8, the bandwidth corresponding to this message size is 30 Mbytes/s, the value we used in our model. A measured value of 150 μs for the latency over the HiPPI link was utilized. The number of links *L* is 4, corresponding to a connectivity of 8 HiPPI links connecting the SMP boxes.

Figures 9, 10 and 11 present the runtime of SWEEP3D on clusters of 2 X 2, 3 X 3 and 4 X 4 Origin 2000 boxes, respectively. The processor configuration inside each SMP box ranges from 2 X 2 to 8 X 8, up to a total of 1024 processors, the largest machine configuration available to us.

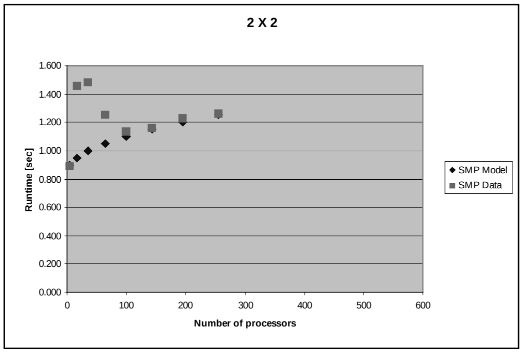

Figure 9. Validation on a 2 X 2 cluster

The model validates well for the 3 X 3 and 4 X 4 cluster configuration, with the exception of one data point in figure 11. As noted by other authors too [9], timing on Origin 2000 machines is a highly idiosyncratic task mainly due to memory placement in its DSM scheme. Memory locality in DSMs directly impacts performance of communication libraries. HiPPI availability is another major source of timing instability, as there is no mechanism for insuring standalone over HiPPI. Contention for HiPPI links from applications running on other SMP boxes in the system can impact the communication time.

We assume that given the shorter communication times in the 2 X 2 case depicted in figure 9, contention for HiPPI plays a major role in the noisier data compared to figure 10 and 11. None of the measurements was done in standalone.

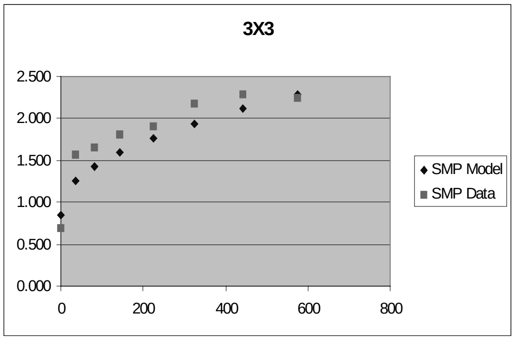

Figure 10. Validation on a 3 X 3 cluster.

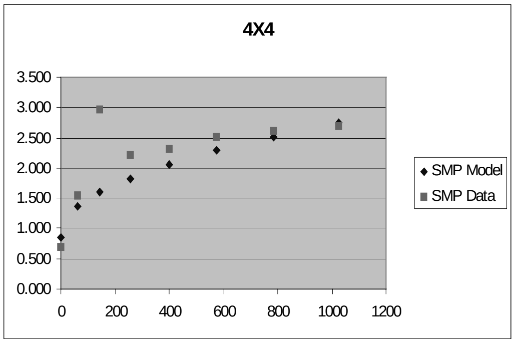

Figure 11. Validation on a 4 X 4 cluster.

6. Conclusions

We proposed a closed-form analytical model for the performance of wavefront algorithms on clusters of SMPs. The model represents a generalization of a previously proposed model applicable to MPP architectures only.

We validated the model on a cluster of Origin 2000 machines, up to a total of 1024 processors. The data supports the validity of the model for all cluster configurations.

The lower-dimensionality of the network topology in a cluster of SMPs, compared to the network in an MPP has a profound impact on the communication performance in the wavefront applications. The model we proposed and validated shows that the impact is not only due to communication parameter changes (as in the different values for the latency and bandwidth across SMP boxes compared to the values inside a SMP box), but more importantly in communication pattern changes. We are not aware of any other performance model of a full application that exposes the algorithmic and performance changes in the application as a result of modifications at the parallel architecture level.

In future work we plan on applying the model to predict performance of very-large scale computations using wavefront algorithms taken from the ASCI workload on Tera-scale architectures organized as clusters of SMPs and analyze and contrast their performance to that of the same applications running on MPP parallel architectures. Such studies can offer significant insight as point design studies for the architecture of parallel systems.

7. References

1. A. Hoisie, O. Lubeck and H. Wasserman, “Performance Analysis of Wavefront Algorithms on Very-Large Scale Distributed Systems”, Lecture Notes in Control and Information Sciences, Springer, Vol 249 page 171, 1999.
2. A. Hoisie, O. Lubeck and H. Wasserman, “Scalability Analysis of Multidimensional Wavefront Algorithms on Large-Scale SMP Clusters”, Proc. Of Frontiers 1999, page 4, February 1999.
3. K. R. Koch, R. S. Baker and R. E. Alcouffe, "Solution of the First-Order Form of the 3-D Discrete Ordinates Equation on a Massively Parallel Processor," Trans. of the Amer. Nuc. Soc., 65, 198, 1992.
4. W. D. Joubert, T. Oppe, R. Janardhan, and W. Dearholt, "Fully Parallel Global M/ILU Preconditioning for 3-D Structured Problems," LANL Technical Report, LAUR 98-2259.
5. J. Qin and T. Chan, “Performance Analysis in Parallel Triangular Solve,” IEEE Second International Conference on Algorithms & Architectures for Parallel Processing, pp 405-412, June 1996.
6. M. T. Heath and C. H. Romine, “Parallel Solution of Triangular Systems on Distributed Memory Multiprocessors,” SIAM J. Sci. Statist. Comput. Vol. 9, No. 3, May 1988.
7. R. F. Van der Wijngaart, S. R. Sarukkai, and P. Mehra, “Analysis and Optimization of Software Pipeline Performance on MIMD Parallel Computers,” Technical Report NAS-97-003, NASA Ames Research Center, Moffett Field, CA, February, 1997.
8. www.hippi.org
9. G. R. Luecke, B. Raffin and J. J. Coyle, “Comparing the Communication Performance and Scalability of a SGI Origin 2000, a Cluster of Origin 2000's and a Cray T3E-1200 using SHMEM and MPI Routines”, PEMCS, October, 1999.